

Models and heuristics for dynamic revenue optimization in road transport

Citation for published version (APA):

van de Klundert, J., & Otten, B. (2007). *Models and heuristics for dynamic revenue optimization in road transport*. METEOR, Maastricht University School of Business and Economics. METEOR Research Memorandum No. 036 <https://doi.org/10.26481/umamet.2007036>

Document status and date:

Published: 01/01/2007

DOI:

[10.26481/umamet.2007036](https://doi.org/10.26481/umamet.2007036)

Document Version:

Publisher's PDF, also known as Version of record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.umlib.nl/taverne-license

Take down policy

If you believe that this document breaches copyright please contact us at:

repository@maastrichtuniversity.nl

providing details and we will investigate your claim.

Joris van de Klundert, Bernhard Otten

Models and Heuristics for Dynamic Revenue
Optimization in Road Transport

RM/07/036

JEL code: C61



Maastricht research school of **E**conomics
of **T**Echnology and **O**Rganizations

Universiteit Maastricht
Faculty of Economics and Business Administration
P.O. Box 616
NL - 6200 MD Maastricht

phone : ++31 43 388 3830
fax : ++31 43 388 4873

Models and Heuristics for Dynamic Revenue Optimization in Road Transport

Joris van de Klundert, Bernhard Otten

Corresponding author: Joris van de Klundert
Department of Quantitative Economics
Faculty of Economics and Business Administration
Universiteit Maastricht
Po Box 616 6200 MD, Maastricht
The Netherlands

jj.vandeklundert@ke.unimaas.nl
+31 43 3883281

September 6, 2007

Abstract

In this document we consider an optimization problem which origins from freight exchanges. The aim is to increase the utilization of already scheduled road transportation activities by accepting extra loads. Such utilization increase reduces costs for the freight sender, increases revenue for the the transportation provider, and reduces congestion and pollution for the society at large.

We explore various models which capture different real life problem settings. Our explorations regard finding good solutions for off line, on line, and on the line versions of the problems. We link the problem with revenue management literature and resource allocation literature, and report computational results on proposed approximation algorithms.

keywords: Transportation, On line algorithms, Models, Heuristics

1 Introduction

The European Environmental Agency (epaedia.eea.eu) reports that the average weight utilization of trucks in Europe is below 50 percent. This may be partly due to the fact that trucks are fully loaded in terms of floor space or volume before reaching their weight limits, but apparently no data on floor utilization or volume utilization are reliably collected. Nevertheless, Samuelsson and Tilanus (1999) provide estimates on various measures for capacity utilization in regional less than truckload (LTL) distribution which confirm that unused capacity is easily in the same order. In view of the enormous value destruction that this unused capacity entails, by under utilization of business assets, by causing congestion, wasting energy resources, and polluting the environment, it is of great interest to the European society, that the transportation capacity utilization increases. This interest perfectly coincides with the interest of transportation companies themselves who are struggling to stay profitable. In The Netherlands for instance, the average return on assets for transportation companies over the years 2000-2005 is negative. In other words, the sector is truly destructing value. Better use of transportation capacity is required to restore the profitability of this sector which is vital to the economy. Finally it is in the interest of consumers as well, since better use of transportation capacity leads to lower costs, and subsequently to lower prices.

The aforementioned observations are not new to policy makers or the sector itself, and many projects have been undertaken to increase capacity usage. One possibility to do so stems from an initiative in the pre internet era, where in France a teletext based Minitel system was used to exchange capacity and truck load among transportation companies. This initiative is the predecessor of the current internet service www.teleroute.com which, like various others, allows various services based on the principle of offering spare capacity and/or load to be transported. In this sequel we call such information providing sites electronic freight exchanges (EFE's).

The basic idea of an EFE is simple. Transportation companies can announce to have loads to be transported, specified by a pick up point and date, a drop off point and date, and parameters such as weight, size et cetera. Conversely, transportation companies who have a trip scheduled and still have spare capacity, announce their trip, giving starting point and drop off point, and the available capacity. Transportation companies who have a load to transport which doesn't justify sending an entire truck can now look for trips with spare capacity which can take their load. Conversely transportation companies with scheduled trips and spare capacity can look for loads that fit their trip. Of course, whenever two transportation companies engage in an agreement they settle on a price for the provided transportation service. In real life, this price may be subject to negotiations which take place in a non public setting, and hence the price is not always published by the EFE. Obviously, a price which is in the interest of both parties involved can often be found since costs are lower when the transports are combined. This statement is confirmed by the fact millions of transaction take place at EFE's per year, and this volume is growing rapidly.

In the remainder of this paper we consider the position of a transportation company which has scheduled a trip defining its pick up and drop off point and remaining capacity and who is seeking extra loads so as to maximize the sum of the revenues from carrying loads. This situation applies in particular to transportation companies who recurringly (e.g. daily) operate fixed origin-destinations trips, and seek for loads on the corresponding route. We introduce the following notation. The set of pick up and drop off points is denoted by $\{c_0, c_1, \dots, c_n\}$ where c_0 is called the origin, and c_n the destination. Further the truck has a capacity C , and there is a set of items $1, 2, \dots, m$, with

- $b_i, i = 1, \dots, m, b_i \in \{c_0, c_1, \dots, c_n\}$ the pick up location of load i ,
- $e_i, i = 1, \dots, m, e_i \in \{c_0, c_1, \dots, c_n\}$ the drop off location of load i ,
- $v_i, i = 1, \dots, m, v_i$ integer, the volume of load i ,
- $r_i, i = 1, \dots, m, r_i$ integer, the revenue of load i ,

This input specification tacitly implies that the 3-dimensional truck capacity, which may also have weight and other restrictions, can be modelled by a single dimensional volume parameter. This single dimension can be the number of pallets that fit in the truck, but is in general an oversimplification of reality. However, our model serves as a basic model in this area which, to the best of our knowledge, hasn't received attention in scientific literature thus far. We refer to this problem as *Truck Load Optimization* problem, or simply *TLO*.

A *solution* to TLO is specified as a subset $S \subseteq \{1, \dots, m\}$ of the set of all items. The *value* $r(S)$ of *solution* S is defined as $r(S) = \sum_{i \in S} r_i$. A solution is called *feasible* if at each point along the trip the sum of the volumes of the loads that are in the truck simultaneously doesn't exceed the truck capacity C . More formally, a solution S is called feasible if

$$\sum_{i \in S: b_i \leq c_j < e_i} v_i \leq C, j = 0, \dots, n-1.$$

We consider three versions of the problem:

1. Classical or *off line* TLO, where all inputs are given on beforehand, and the task is to find a feasible solution of maximum value.
2. *On line* TLO, where the $\{c_0, c_1, \dots, c_n\}$ are given on beforehand, but the loads are given one by one. The task is to build a feasible solution by accepting or rejecting loads when they are offered, so as to maximize the value of the accepted loads.
3. *On the line* TLO, where the $\{c_0, c_1, \dots, c_n\}$ are given on beforehand, but the loads are given as the truck makes his trip. Whenever the truck arrives at a next location $c_j, j = 0, \dots, n$, the loads $i, i = 1, \dots, m$ such that $b_i = c_j$ are offered simultaneously. The task is to build a feasible solution by accepting or rejecting loads when they are offered, so as to maximize the value of the accepted loads.

It will become clear in subsequent sections that TLO is closely related to several classical combinatorial problems such as the Knapsack problem and the Minimum cost flow problem. In this paper we investigate how these fundamental combinatorial properties can be used to design effective heuristics for the various versions of TLO.

TLO is related to a variety of problems which have recently received considerable attention from a complexity and approximability viewpoint. We mention the work of Bar Noy et al. (2001) who consider a quiet general resource allocation model, which applies to parallel machine scheduling and bandwidth allocation in telecom networks among other problems. The bandwidth allocation problem is the problem of allocating bandwidth between two servers in a communication network over time. The bandwidth capacity must be allocated to sessions, each of which have a starting end ending time, require a certain bandwidth, and have a certain profit. The task is to select a set of sessions so that at each moment in time the used bandwidth doesn't exceed the available capacity, and so that the sum of the profits of the set of selected sessions is maximum. This problem can be seen to equivalent to TLO as follows. Imagine that the TLO truck travels at constant speed from the origin to its destination, starting at time zero, and arriving 'sufficiently late'. Now, each of the starting times of the sessions can be represented by a pick up point between the origin and the destination, and each ending time by a drop off point (thus 'sufficiently late' means after the last drop off point has been encountered). Then, sessions can be interpreted as loads with profits, volume requirements, and pick up and drop off points. A $\frac{1}{3}$ approximation algorithm for TLO is implied by the work of Calinescu et al (2002) to which we refer for further references on related work in the bandwidth allocation setting, such as resource constrained scheduling, call admission control, and the resource allocation problem, for problems that are akin or equivalent to off line TLO (see also (Bansal et al. 2006)).

The results of Bar Noy et al. (2001) also yield a $\frac{1}{3}$ approximation algorithm and provides a thorough overview of related work in approximation and complexity of closely related scheduling problems. The version of TLO in which there is unit capacity can easily be cast as a single machine scheduling problem. The single machine has unit capacity, jobs require unit capacity, and have release dates r_j (here we briefly change notation to the classical scheduling notation), deadline d_j and processing time p_j . Interpreted as such, TLO is a special case where $p_j = d_j - r_j$. The single machine version of this special case is considered by Arkin and Silverberg (1987) who provide a polynomial algorithm (which is equivalent to the min cost flow model presented below.) Woeginger (1994) considers on line versions of this problem, which is in general known not to allow a finite worst case guarantee for on line algorithms. However he considers special cases and presents on line algorithms with a worst case ratio of four. The multiple machine generalizations of these scheduling problems still require all jobs to have a volume of one, that is each of the jobs can and must be processed using only one of the machines. The version of TLO where the jobs have unit volume is demonstrated to be polynomially solvable below. We

refer to (Chuzhoy et al. 2001) and (Erlebach and Spieksma 2003) for work on hard generalizations of such scheduling problems.

TLO can also be considered as a revenue management problem. Revenue management problems are mainly investigated in consumer markets such as passenger airlines or hotels. In such settings, the revenue management problem consists in offering capacity for the right prices and accepting the right customer requests, so as to maximize total revenue. In the airline industry, a basic model is the origin destination (OD) pair problem, where an airplane with a fixed number of seats travels from an origin to a destination, and seeks to select customers who travel between these two locations. (Multiple location versions of this problem are known as single resource models.) Customers are segmented into various classes, each with different arrival rates and prices. The problem is then to devise acceptance criteria for requests from each of the classes. Various model variations exist, see e.g. Chapter 2 of Talluri and Van Ryzin (2005) for an excellent overview.

In the revenue management literature acceptance criteria are known as ‘controls’, which can either be static or dynamic. An important difference between these models and TLO is that in the TLO the price is given by the customer, whereas in the revenue management problem the prices are often set by the vendor.

Network versions of revenue management problems also exist. In such network versions, passengers travel from various origins to various destinations, perhaps sharing flights as they go. A simple one dimensional analogue of such a problem arises in the hotel industry, where the locations are the days, and hotel guests book rooms for sets of consecutive days (Talluri and Van Ryzin 2005, Chapter 3) for an overview of finding sets of controls, either optimally or heuristically. The setting from the hotel industry is quite close to TLO, as it can also be modeled using the same line graph sketched above. However, the problem differs from TLO in several respects. Typically, hotels clearly distinguish customer segments (business, leisure, wholesale et cetera). Moreover, capacity is divided over rooms of several types, and for each segment prices are in principle per room per night. This price structure is much more specific than what TLO assumes. Moreover, in TLO capacity is continuous, and loads have a size from a continuous domain.

In this paper, we only partly build on the work on bandwidth allocation and/or revenue management. From the revenue management literature we do borrow the concept of controls. In addition we apply combinatorial heuristics and techniques as they are also used in the work on bandwidth allocation. However, our focus will be on solving the aforementioned basic versions of TLO using its relationship with the Knapsack problem and the min cost flow problem. In Section 2 we show that the special case where $v_i = K, i = 1, \dots, m$ can be modelled as min cost flow problem and hence be solved in polynomial time. We use this min cost flow formulation as a base for one of several approximation algorithms for which we report worst case analysis and computational results.

The other approximation algorithms will be 'control' like algorithms inspired by approximation algorithms for the knapsack problem. In Section 3 we consider On line TLO, where we are primarily concerned with exploring on line algorithms which perform well on a set of generated test instances. The thus developed ideas, as well as others, will subsequently be adapted to fit the case of On the line LTO in section 4. Section 5 contains conclusions and suggestions for further research.

2 Off line TLO

2.1 Complexity

Off line TLO will serve as the basic model and point of reference in this paper. Therefore we start in this subsection by analyzing its complexity. As is well known (Bar Noy et al. 2001) even the version where all loads have the origin as their pick up city, and the destination as the drop off city is NP-complete. This can be seen by noticing that this special case of TLO is simply a restatement of the knapsack problem. Let it be noted however that the Knapsack problem is only weakly NP-complete. Bansal et al. (2006) provide a quasi PTAS for TLO (UFP on a line graph, the Resource Allocation Problem) and conclude that it is unlikely to be APX hard. Whether TLO is strongly NP-complete appears to be open.

Let us now consider the relaxation of TLO where loads can be partially selected. The following formulation as a min cost flow problem in a directed graph $D(N, A)$ (see also ?? is from (Gupta 1985, Ahuja et al. 1993). A closely related graph model is proposed by Arkin and Silverberg (1987).

There is a node N_i for each load $i, i = 1, \dots, m$. This node has a supply of v_i . Further there are nodes for each of the locations on the path that the truck travels, nodes c_0, \dots, c_n . There are arcs $(c_j, c_{j+1}), j = 0, \dots, n-1$, with costs 0 and capacity C , denoted by $[0, C]$. Moreover, there are arcs $(n_i, b_i), i = 1, \dots, m$, that is from a load node n_i , to its pick up location b_i , also with costs 0 and capacity C . Finally there are arcs $(n_i, e_i), i = 1, \dots, m$, that is from a load node n_i to its drop off location e_i , with capacity C and cost $\frac{r_i}{v_i}$, the per unit revenue of item i . The nodes $c_j, j = 1, \dots, m$, have demand equal to $\sum_{i=1}^m v_i |e_i = c_j$.

Now let F be a feasible flow in $D(N, A)$. Obviously it is a collection of path flows, where each path runs from a load node $n_i, i = 1, \dots, m$ to the corresponding drop off point e_i . Notice also that once a path to e_i contains a node c_k for some $k \in 1, \dots, k$ than it per force continues by visiting $c_{k+1}, c_{k+2}, \dots, e_i$. Hence from each load node n_i , F can contain two paths, the long path (l_i, b_i, \dots, e_i) of which each arc has cost zero, and the short path (l_i, e_i) which consists of one arc of cost $\frac{r_i}{v_i}$. Naturally a min cost flow solutions seeks to reduce the costs, preferring not to use the short paths. However, the long paths share the arcs of the type (c_j, c_{j+1}) which have a capacity upperbound of C .

Now, for any feasible flow F , and for $i = 1, \dots, m$ let $f_i, 0 \leq f_i \leq v_i$ be the amount of flow which travels along the short paths. Then, F has value $\sum_{i=1}^m f_i \times \frac{r_i}{v_i}$ or equivalently $\sum_{i=1}^m \frac{f_i}{v_i} \times r_i$. The flow can be easily interpreted in terms of the relaxed version of TLO. Accepting from each item i a part by the size of $v_i - f_i$ yields a feasible solution, that is all these items fit into the truck capacity C between every pair of consecutive locations $c_j, c_{j+1}, j = 0, \dots, n-1$. Moreover, in the relaxed version of TLO this solution has total revenue $\sum_{i=1}^m \frac{(v_i - f_i)}{v_i} \times r_i = \sum_{i=1}^m (r_i - \frac{(f_i)}{v_i} \times r_i)$. We conclude that the value of the flow F corresponds to the TLO revenue missed by (partially) unaccepted loads. Thus minimizing the value of the flow F corresponds to minimizing the opportunity cost of unaccepted load. Thus, a min cost flow solution maximizes total revenue in the corresponding relaxed TLO problem.

It is well known (Ahuja et al. 1993) that min cost flow problems can be solved in polynomial time, and that there always exists an integral optimal solution, which can also be found in polynomial time. Now let us address the implications of these properties of relaxed TLO to TLO. In TLO, the flow which corresponds to a load is not allowed to be split, that is to be send to its drop off point along two paths. (Hence, we have reestablished its interpretation as an unsplittable flow problem (Bansal et al. 2006)). Now consider the special case where all loads have equal volume, that is for some $K \leq C, K \in N$, and for all $i, i = 1, \dots, m$ $v_i = K$. We refer to this special case by TLO- K . Normalizing the v_i by setting them to 1, and letting $C = \lfloor \frac{C}{K} \rfloor$, we obtain an equivalent TLO-1. Hence, the supply of each load node equals 1, and therefore an integral solution sends it either entirely along the long path, or entirely along the short path. Since, an optimal integral solution to min cost flow can be found in polynomial time, TLO-1 is polynomially solvable. Thus, we have established the following theorem:

Theorem 1 *Off line TLO- K is polynomially solvable.*

2.2 Basic Approximation Algorithms

In view of the practical application which motivates this paper, we are not primarily interested in solution approaches for off line TLO. Our aim is to derive practical solution methods for on line TLO and for on the line TLO. As a reference we are however interested in the off line optimal solution, and the behavior of several solution approaches in an off line setting. We therefore now consider several basic polynomial approximation algorithms for off line TLO, which can serve as a basis for practical solution approaches in on line TLO and on the line TLO in subsequent sections. The first two approaches will be based on the analogy with the knapsack problem, the latter two on the min cost flow relaxation.

1. **Density Greedy :** The Density Greedy heuristic is a direct translation of the Greedy heuristic for the Knapsack problem. First, we enlist the set of loads in descending order of $\frac{r_i}{v_i}$. Subsequently we process the list in

order, and repeatedly select a load if together with the already selected loads it forms a feasible solution to TLO.

2. **Distance Adjusted Density Greedy :** Obviously, Density Greedy doesn't take into account whether loads make short or long trips. Therefore this second list processing algorithm attempts to take the travelled distance into account, by dividing the density of an item by the distance it has to travel. For load i , let k and l be such that $c_k = b_i$ and $c_l = e_i$. Define (the distance of load i) $d_i = l - k, i = 1, \dots, m$. Then, we enlist the set of loads in descending order of $\frac{r_i}{v_i \times d_i}$. Subsequently we process the list in order, and select a load if together with the already selected items it forms a feasible solution to TLO.
3. **MCFI :** This algorithm considers an optimal solution F to the corresponding min cost flow relaxation, and selects exactly those loads $i, i = 1, \dots, m$ for which $f_i = 0$.
4. **MCFR :** This algorithm extends the MCFI algorithm above, by selecting loads in addition to the items selected in the solution constructed by MCFI if possible. To this purpose, we sort the loads in ascending order by the fraction $\frac{f_i}{v_i}$. Now we process this list, while drawing a random number x_i uniformly in the interval $(0, 1)$ with each load i . We select item j if $x_i > \frac{f_i}{v_i}$. Thus indeed, all loads included in MCFI are included since they have $\frac{f_i}{v_i} = 0 < x_i$, and the other loads are selected with a probability according to the ratio in which they are selected in the solution to the min cost flow relaxation. As a final step of the algorithm, we decided to process the list of not yet selected loads again, in the same order, and simply select them if they can be added to the current solution without rendering it to be infeasible. This algorithm is not deterministic, but better captures the potential of the min cost flow formulation than plain MCFI.

For the knapsack problem, there is a very strong relationship between the solution of the greedy algorithm and the solution of the LP relaxation (or equivalently, the min cost flow relaxation). This relationship however cannot be directly translated to TLO. Therefore the performance of the greedy approaches and the min cost flow based approaches can be quite different.

Despite the fact that the Greedy algorithm for the Knapsack problem has a worst case ratio of $\frac{1}{2}$, one easily shows that the aforementioned greedy approximation algorithms have infinite worst case behavior. The same holds for the min cost flow rounding algorithm MCFI.

Finally, we notice that solving the min cost flow instance, which corresponds to a relaxed instance of TLO, provides an upperbound on the optimal solution value, to be referred to as MCF. We conjecture that the optimal solution of TLO is bounded from below by $\frac{1}{2}$ the optimal value of the LP relaxation, which might lead to developing approximation algorithms with a tighter ratio than the currently best known.

We have tested the basic approximation algorithms on a number of randomly generated sample instances. We generated 16 different sets of 50 instances each. The 16 different sets result from trying two choices for each of the following parameters:

1. Volume of the loads, we either vary between 0 and 20, or between 7 and 13, sampling from a discrete uniform distribution,
2. Number of locations, either 10 or 20. For each load, we randomly select two locations, using a discrete uniform distribution, and (if they are not equal), we let the smaller one be the pick up location, and the larger one be the drop off location. (If they are equal, we retry.)
3. Load $i, i = 1, \dots, m$ has an expected revenue of $d_i * v_i$, that is it's expected revenue equals the travel distance times volume. We randomly generate a density for each load, and multiply by distance times volume. We generate instances either by sampling densities from a continuous uniform distribution with domain $[0.9, 1.1]$, or by using a continuous uniform distribution with domain $[0.5, 1.5]$. In the first set of instances optimality might be harder but errors less costly.
4. The capacity of the truck is either 30 or 100.

The instances are generated in such a way that, roughly speaking, half of the items can be selected. For the instances with 10 locations and capacity 30 we therefore generated 20 loads. For instances with 20 locations and capacity 30, we generated 30 loads. For the instances with capacity 100, we generated 55 loads. Scaling the min cost flow upper bound to 100 percent, we obtain the following results of Table 1.

We conclude that, although none of the approximation algorithms behaves really poor, there is quite a variance. The pure min cost flow rounding approach is the worst for each set of instances. The difference between the min cost flow upper bound and the randomized MCFR is never more than 12 %, and for the larger instances it is within 4 %. The plain greedy approach of DG gives a performance which is quite consistently around 10 % below the upper bound. DADG does much poorer on the smaller instances, but catches up as the instances get larger.

It may appear counter intuitive that DG gives better solutions than DADG. After all, DADG selects items with a higher value density per travel distance. The fact that DG gives better solutions can however be explained as follows. DG selects item with a high value, which favors loads i with large d'_i s, since the instances are generated in such a way that r_i is proportional to d_i . Thus, DG tends to consider loads in order of decreasing travel distance. DADG instead

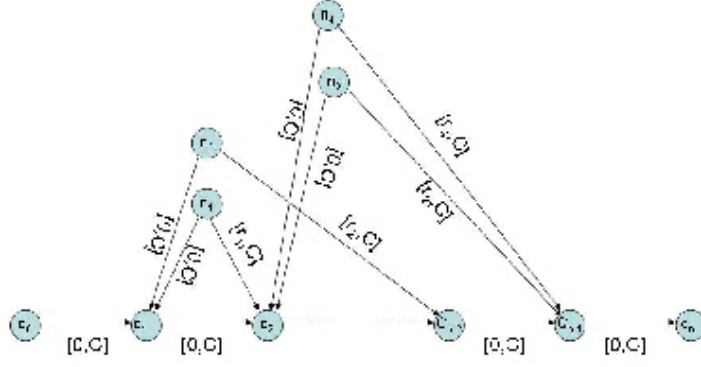


Figure 1: Figure 1: Min Cost Flow Model

<i>Instance parameters</i>	<i>DG</i>	<i>DADG</i>	<i>MCFI</i>	<i>MCFR</i>
V:20; C:30; p:0.9-1.1 w:1-20 n:30	87.92	72.84	63.85	89.89
V:20; C:30; p:0.9-1.1 w:7-13 n:30	84.92	73.32	65.79	90.96
V:20; C:30; p:0.5-1.5 w:1-20 n:30	87.92	72.84	66.59	88.35
V:20; C:30; p:0.5-1.5 w:7-13 n:30	86.66	78.35	74.87	90.86
V:10; C:30; p:0.9-1.1 w:1-20 n:20	87.58	75.06	66.38	90.63
V:10; C:30; p:0.9-1.1 w:7-13 n:20	86.04	78.67	70.96	92.32
V:10; C:30; p:0.5-1.5 w:1-20 n:20	87.32	82.54	71.17	89.68
V:10; C:30; p:0.5-1.5 w:7-13 n:20	85.82	81.12	77.79	92.01
V:20; C:100; p:0.9-1.1 w:1-20 n:55	91.07	84.61	80.07	96.92
V:20; C:100; p:0.9-1.1 w:7-13 n:55	90.34	84.19	83.73	97.25
V:20; C:100; p:0.5-1.5 w:1-20 n:55	92.47	89.61	86.17	97.16
V:20; C:100; p:0.5-1.5 w:7-13 n:55	92.28	89.87	88.87	97.44
V:10; C:100; p:0.9-1.1 w:1-20 n:55	92.65	87.48	82.37	98.34
V:10; C:100; p:0.9-1.1 w:7-13 n:55	89.11	86.86	76.11	96.21
V:10; C:100; p:0.5-1.5 w:1-20 n:55	93.61	92.17	86.50	98.75
V:10; C:100; p:0.5-1.5 w:7-13 n:55	94.20	92.54	90.47	98.53

Table 1: Heuristic Results for off line TLO

considers them strictly in density order. As a result DADG may select high value loads with short distances, and later have no space for loads with lower density but longer distance and therefore higher revenue. DA has an easier time to fill up the capacity it leaves unused and therefore finds a better solution.

Thus, the relative performance of the two appears to be highly dependent on two assumption which are implicit in the generated instances. The first assumption is that revenue is directly proportional to distance. The second assumption is that the number of items is limited such that about half of them can be accepted. Therefore we have dropped each of these assumptions in turn. First we generated instances where revenue was independent of distance. For these instances DADG outperformed DG by a wide margin (see (Otten 2006) for more detailed results). Although these results may be more general from a conceptual viewpoint, prices in road transportation are usually linear in the distance traveled. Therefore, we maintain our first assumption in the remainder. Second we generated instances where the set of loads was much larger. In this case, both algorithms consistently realize a total revenue of about 90 % of the min cost flow upperbound. Such instances are challenging from the viewpoint of computing an exact solution. However, for practical purposes they are less challenging since they tend to be more forgiving. If the list of offered loads is long enough, there is always a load not considered yet that approximately fills up the open spaces along the route.

3 On line TLO

In real life, the problem setting is more challenging than the setting considered in the previous section. Loads are being offered on an EFE, which is a public trading exchange. If one were the only transportation provider, one could simply collect the load information until the moment of departure, solve the off line problem, and select the corresponding items. In reality however, other transportation providers are active on the market as well, and to model this environment, we consider a standard and simple on line optimization model. As soon as a load is offered, one has to accept or reject it. As is often assumed for the on line knapsack problem (Marchetti-Spaccamela and Vercellis 1995), we assume the transportation provider to have a realistic expectation about the number of loads that will in total be offered. This is not equivalent, but similar to a model in which time progresses, and the transportation provider has an expectation about the arrival rate of offered loads.

The on line problem setting is much more akin to the revenue management problems in airline industry which is mentioned in the introduction. Moreover, it is a generalization of the on line knapsack problem, as for instance studied by Marchetti-Spaccamela and Vercellis (1995), who show that any on line algorithm for the knapsack problem has infinite worst case behavior.

The on line algorithms we propose and computationally compare in this section are based on our findings in the previous paragraph, and ideas which

have proven to be useful in related contexts. For example, from the revenue management literature, we borrow the idea of using (dynamic) controls. Further, we use Monte Carlo simulation based approaches.

1. **Fit** : The first approach is to accept any load that fits. This is of course a simple and extremely Greedy approach, and any intelligently designed algorithm should outperform this approach.
2. **TH** : The second approach is to accept any load i for which the density $\frac{r_i}{v_i}$ exceeds a threshold value T .
3. **TLC** : In this approach, the threshold decreases (linearly) in steps of equal size from value Top to value $Base$, by setting the threshold for the i -th load to be $Base + (Top - Base) \times (1 - \frac{(i-1)}{m-1})$. Finding the best choice of parameters Top and $Base$, $Top \geq Base$ is part of the computational experiments.
4. **TED** : In this approach, the threshold decreases (exponentially) in steps of diminishing size from value Top towards value $Base$. Analogous to the previous approach, we therefore set the threshold for the i -th load to be $Base + (Top - Base) \times (e^{-\phi \frac{i}{m}})$. Finding the best choice of parameters ϕ , Top and $Base$, $Top \geq Base$ is again part of the computational experiments.
5. **MLD** : In this approach, we use the min cost flow formulation as follows. We generate 50 instances which consist of the already revealed instance plus a set of randomly generated subsequent loads. For each of the thus created random instances, we check whether the current item is (perhaps fractionally) selected in a best solution which uses the already constructed on line solution as a starting point. This gives us 50 fractions for load i which add up to say F_i . Moreover, we calculate the sum of all fractions of selected loads in the optimal solutions of the 50 instances. Let S_i be this sum. We accept load i if $F_i > l \times \frac{S_i}{m}$, where l is again a parameter which is to be determined in the computational experiments.
6. **MED** : This approach is the same as the MCF constant threshold approach, only now the threshold decreases exponentially (as in the exponential dynamic control threshold).

In order to compare the different algorithms, we report on a relatively hard set of instances, consisting of 20 locations and a truck capacity of 30. Insight in the effectiveness of especially the dynamic control algorithms is obtained by studying instances with 50, 100 and 200 loads. This is an order of magnitude larger than in the previous section, and we will reconsider this issue when discussing the results. Scaling, the results by setting the value of the MCF upperbound, we obtain the results of Table 2. We also present the underlying absolute results in Table 3, since they are insightful in itself.

The table displays the results for three sets of instances. The first set consist of instances of 50 loads, where the other parameters are as before. The second

and third set consist of instances of 100, resp. 200 loads, with the same instance parameters. The latter instances which consists of longer lists of loads offer more room to optimize for selective algorithms. Again, per line of the table, the results are averages over 50 instances.

The first column gives the value of the myopic FIT approach, which is to accept any load that fits, as a percentage of the value of the min cost flow relaxation. Certainly, each of the other algorithms should do better than FIT. The other columns give the values of the other algorithms, again as a percentage of the min cost flow lowerbound. The last column gives the value of the best off line algorithm MCFR.

Each of the algorithms with which the results in the columns two to six are obtained involves parameters. Parameter values are heuristically selected so as to optimize the average score over all sets of instances. The second, third and fourth column are obtained by selection rules TH, TH-LC, and TH-DC, which select loads based on their density. The latter two are dynamic, TH-DC being the most general and flexible. Hence one would expect TH-DC to outperform TH-LC, which in turn should do better than TH. Overall this indeed the case, but for specific sets of instances, it need not be the case. The parameters of TH-LC appear to favor smaller instances, whereas the parameters of TH-DC appear to favor larger instances.

Notice also that for the instances for which the differences in density of the loads are small (that is where p ranges from 0.9 to 1.1), TH has almost identical performance to the rudimentary FIT, as does TH-DC.

The min cost flow based approaches MLD and MED don't perform well. Despite their use of the combinatorial structure of the problem, and monte carlo analysis, their performance is not consistently better than the performance of the simple density threshold strategy TH.

Let us also notice that the poorest strategy, FIT, results in a revenue percentage of the off line MCF upperbound of less than 68 % on average. Taking into account that the MCF upperbound also doesn't load the truck fully, we see that truck utilizations of 50 %, as reported in the introduction, are probable. Let it be noted that percentages differences in solution value are differences in revenue which can be obtained with relatively smaller differences in costs. Hence, from the viewpoint of the transportation company, the results indicate that there is quite a potential in improving operational results by using more advanced acceptance criteria. The absolute results as presented in Table 3 underline these conclusions once again.

4 On the line TLO

The On the line TLO version of the TLO problem is perhaps somewhat futuristic in a road transport setting. It assumes transportation companies to consider accepting loads while moving towards their destination. Therefore, this is perhaps more fitting for transport by ship. However, mobile technology makes on the line TLO a realistic model, whether on the road or on water, and we explore

<i>Instance parameters</i>			<i>FIT</i>	<i>TH</i>	<i>TLC</i>	<i>TDC</i>	<i>MLD</i>	<i>MED</i>	<i>Offline</i>
0.9-1.1	1-20	50	72.51	72.51	77.31	73.43	72.51	72.51	90.96
0.9-1.1	7-13	50	71.40	71.40	76.68	73.22	71.40	71.40	91.99
0.5-1.5	1-20	50	61.81	66.26	67.48	72.24	66.41	69.48	91.72
0.5-1.5	7-13	50	59.73	69.31	70.06	72.90	68.26	71.56	92.07
0.9-1.1	1-20	100	75.78	75.78	77.00	78.22	75.78	75.78	93.21
0.9-1.1	7-13	100	74.48	74.48	77.83	78.62	74.48	74.48	92.76
0.5-1.5	1-20	100	60.53	68.26	72.84	72.47	66.85	70.51	91.99
0.5-1.5	7-13	100	60.45	67.27	72.84	73.68	70.06	72.28	91.36
0.9-1.1	1-20	200	78.66	78.66	78.66	80.67	78.66	78.66	93.28
0.9-1.1	7-13	200	77.18	77.35	77.35	80.20	77.35	77.35	93.29
0.5-1.5	1-20	200	61.81	72.57	67.45	77.30	69.69	71.39	92.26
0.5-1.5	7-13	:200	58.98	72.53	66.80	76.95	70.05	72.26	91.80

Table 2: Relative results of on line TLO heuristics

<i>Instance parameters</i>			<i>FIT</i>	<i>TH</i>	<i>TLC</i>	<i>TDC</i>	<i>MLD</i>	<i>MED</i>	<i>Offline</i>	<i>MCF</i>
0.9-1.1	1-20	50	393	393	419	398	393	393	493	542
0.9-1.1	7-13	50	392	392	421	402	392	392	505	549
0.5-1.5	1-20	50	403	432	440	471	433	453	598	652
0.5-1.5	7-13	50	399	463	468	487	456	478	615	668
0.9-1.1	1-20	100	435	435	442	449	435	435	535	574
0.9-1.1	7-13	100	432	434	434	456	432	432	538	580
0.5-1.5	1-20	100	431	490	486	516	476	502	655	712
0.5-1.5	7-13	100	434	501	483	529	503	519	656	718
0.9-1.1	1-20	200	468	468	468	480	468	468	555	595
0.9-1.1	7-13	200	460	461	461	478	461	461	556	596
0.5-1.5	1-20	200	471	553	514	589	531	544	703	762
0.5-1.5	7-13	200	453	557	513	591	538	555	705	768

Table 3: Absolute results of on line TLO heuristics

the behavior of several heuristics in this section.

On the line problems have received some attention in the literature. In particular, TSP on the line has been investigated as a polynomially solvable special case (Feuerstein and Stougie 2001). Moreover, several (multi) server problems with specific underlying transportation structures have been investigated, leading to problems such as the CNN problem (Koutsoupias and Taylor 2004). However, research on the CNN problem, and other single server problems is not directly related to the on the line model discussed in this section.

All of the approaches explored in this section can be viewed as on the line versions of the on line algorithms in the previous section.

1. **Fit :** We travel and consider the loads in order of their starting location. Per starting location, we consider them in random order and accept any load that fits. This is of course a simple and extremely Greedy approach, and any intelligently designed algorithm should outperform this approach.
2. **Greedy Fit :** The second approach differs from the previous one as follows. Per starting location, all loads are sorted in decreasing order of their weights.
3. **Constant Control :** Per location, we accept any load i for which the density $\frac{r_i}{v_i}$ exceeds a threshold value T . per location, we consider the loads in decreasing order of their density. Threshold parameter T is determined in the computational experiments.
4. **Linear Dynamic Control :** In this approach, the threshold increases in steps of equal size from value Top to value $Base$, by setting the threshold for the j -th location to be $Base + (Top - Base) \times (1 - \frac{(j-1)}{n-1})$. Finding the best choice of parameters Top and $Base$, $Top \geq Base$ is part of the computational experiments.
5. **Exponential Dynamic Control :** In this approach, the threshold increases in steps of diminishing size from value Top towards value $Base$. Analogous to the previous approach, we therefore set the threshold for the j -th location to be $Base + (Top - Base) \times (e^{-\phi \frac{j}{n}})$. Finding the best choice of parameters ϕ , Top and $Base$, $Top \geq Base$ is again part of the computational experiments.
6. **MCF constant threshold :** In this approach, we use the min cost flow formulation as follows. We proceed location by location, and in each iteration we therefore simultaneously consider all loads with a common starting location, say location j . We complete the current instance in 50 different ways by generating random loads for subsequent starting locations. over all of the thus created random instances, we sum for each load i with $b_i = j$ the fractions by which it is selected in the corresponding solutions, which extended the solution already constructed up to and including location $j - 1$. This gives us 50 fractions for load i which add up to say F_i . As before, we calculate the sum of all fractions of selected

loads in the optimal solutions of the 50 instances. Let S_i be this sum. We accept load i if $F_i > l \times \frac{S_i}{m}$, where l is again a parameter which is to be determined in the computational experiments.

7. **MCF exponential threshold :** This approach is the same as the MCF constant threshold approach, only now the threshold decreases exponentially as in the exponential dynamic control threshold.

The results in tables 4 and 5 are obtained using the same instances as the results of Table 2 and 3 in the previous section. Hence a first and remarkable observation, is that the results on the on the line are considerably better than the results on the on line instances, and indeed quite close to the off line results, especially for the instance with small variation in density. This can be explained by the fact that the on the line version possesses considerably more structure than the on line instance. Firstly, loads are offered in groups, rather than one by one. Secondly, the groups are ordered in increasing index of their pick up location, or chronologically. Apparently the two of them combined, make the instances almost as easy (or hard) to solve heuristically as off line instances.

Again we notice that the MCF based algorithms don't outperform the simple threshold approach of the second column. In general, one might even say that none of the approaches performs significantly better than the straightforward FIT algorithm on the instances with low density variation. For instances with higher density variation however, we see that the dynamic (exponential) threshold approach performs consistently better than the others. For completeness, we present the absolute results below.

5 Conclusions and further research

Weight utilization of trucks in Europe is reported to be below 50 %. Although weight utilization is not an objective in itself, many compelling reasons can be listed to increase the utilization rate of truck capacity. In this paper we study the topic of maximizing revenue from road transport capacity for less than truck load transport. Noticing that revenues for transport are often in euro per ton-kilometers, revenue maximizing is closely related to weight utilization, and it is the natural incentive for the sector itself.

This paper briefly explores the problem from a theoretical viewpoint, and although there is much work done in related areas such as bandwidth allocation and revenue management, some important questions appear to be still open. First of all, it is not known whether the problem is strongly NP-complete. Second, although the problem is unlikely to be APX-hard, it's approximability status is open. Thirdly, there is no tight bound on the integrality gap, where the LP-relaxation is a simple min cost flow problem. We expect that a deeper understanding of the relationship with the LP-relaxation will lead to improved approximation algorithms as well. In addition, no performance bounds exist for the on the line version of the problem, which appears to be much easier to solve approximately than the on line version of the problem.

<i>Instance parameters</i>			<i>FIT</i>	<i>TH</i>	<i>TLC</i>	<i>TDC</i>	<i>MLD</i>	<i>MED</i>	<i>Offline</i>
0.9-1.1	1-20	50	87.45	87.45	87.45	87.45	87.45	88.75	90.96
0.9-1.1	7-13	50	85.43	85.43	85.43	85.43	87.43	87.80	91.99
0.5-1.5	1-20	50	77.61	81.75	81.44	84.05	80.37	77.76	91.72
0.5-1.5	7-13	50	80.41	83.47	82.90	85.72	81.99	80.07	92.07
0.9-1.1	1-20	100	90.22	90.22	90.24	90.66	90.89	90.76	93.21
0.9-1.1	7-13	100	88.56	88.56	88.56	88.71	90.22	90.51	92.76
0.5-1.5	1-20	100	81.17	84.03	83.60	87.03	82.46	81.63	91.99
0.5-1.5	7-13	100	83.01	85.16	84.70	87.91	83.38	83.07	91.36
0.9-1.1	1-20	200	92.30	92.30	92.30	92.54	91.18	91.86	93.28
0.9-1.1	7-13	200	90.49	90.49	90.49	90.57	91.00	90.64	93.29
0.5-1.5	1-20	200	85.45	87.44	86.60	88.97	84.78	83.97	92.26
0.5-1.5	7-13	200	88.54	88.66	88.54	90.33	86.43	86.85	91.80

Table 4: Relative results of on the line TLO heuristics

<i>Instance parameters</i>			<i>FIT</i>	<i>TH</i>	<i>TLC</i>	<i>TDC</i>	<i>MLD</i>	<i>MED</i>	<i>Offline</i>	<i>MCF</i>
0.9-1.1	1-20	50	474	474	474	474	474	481	493	542
0.9-1.1	7-13	50	469	469	469	469	480	482	505	549
0.5-1.5	1-20	50	506	533	531	548	524	507	598	652
0.5-1.5	7-13	50	537	558	554	573	548	535	615	668
0.9-1.1	1-20	100	518	518	518	520	522	521	535	574
0.9-1.1	7-13	100	514	514	514	515	523	525	538	580
0.5-1.5	1-20	100	578	598	595	620	587	581	655	712
0.5-1.5	7-13	100	596	611	608	631	599	596	656	718
0.9-1.1	1-20	200	549	549	549	551	543	547	555	595
0.9-1.1	7-13	200	539	539	539	540	542	540	556	596
0.5-1.5	1-20	200	651	666	660	678	646	640	703	762
0.5-1.5	7-13	200	680	681	680	694	664	667	705	768

Table 5: Absolute results of on the line TLO heuristics

In our computational experiments we have constructed algorithms which make use of basis properties of the problem, among which is its interpretation as a min cost flow problem when the integrality constraints are relaxed. Moreover, we have studied dynamic threshold strategies as they are used in revenue management, and which are closely related to the greedy algorithm for the Knapsack problem. Our results indicate that we have not been able to leverage the combinatorial properties of the min cost flow based approaches, as they are outperformed by the threshold strategies. Further work in this direction may yield better results.

Our computational results indicate that the straightforward on line setting we consider for on line TLO makes it very difficult to obtain good solutions. The on the line setting, allows for much better revenue and capacity utilization. Certainly much work can be done on a further understanding of these problems, and deriving better solution approaches. Both settings are however stylized versions of reality, in which loads are traded in an EFE, a public market place. In order to further develop solution approaches that are practically relevant, less stylized models can be considered. In particular a context in which several transportation providers are active is relevant, as are contexts in which prices are more dynamic (increase over time for unaccepted loads). Moreover, many transportation companies operate transportation networks, in which transportation lines exist, but are not considered in isolation.

Finally, we conclude that the differences in outcomes for on line and on the line instances reveal that the mechanisms and technology used by the EFE's strongly influence the results as well. Hence the issue of setting up mechanisms and rules for EFE's which make maximum use of state of the art technology is a promising direction for future research as well.

References

- [1] Arkin E. M., Silverberg E.B.,(1987. Scheduling jobs with fixed start and end times, *Discrete Applied Mathematics* 18(1), 1-8,
- [2] Ahuja, Ravindra K., Magnanti, Thomas L., Orlin, James B., 1993. *Network Flows, Theory, Algorithms, and Applications*, Prentice Hall.
- [3] Bansal, Nikhil, Chakrabarti, Amit, Epstein, Amir, Schieber, Baruch, 2006. *A quasi-PTAS for unsplittable flow on line graphs*, Proceedings of the 38th annual ACM symposium on Theory of computing, 721–729,
- [4] Bar-Noy, Amotz, Bar-Yehuda, Reuven, Freund, Ari, Naor, Joseph, Schieber Baruch, 2001. *A unified approach to approximating resource allocation and scheduling*, Journal of the ACM, 48 (5), 1069-1090,

- [5] Calinescu, Gruia, Chakrabarti, Amit, Karloff, Howard J., Rabani Yuval, 2002. *Improved approximation algorithms for resource allocation*, Proceedings of the 9th International IPCO Conference on Integer Programming and Combinatorial Optimization, 401-414,
- [6] Chuzhoy, J., Ostrovsky, R., Rabani, Y., 2001. *Approximation algorithms for the job interval selection problem and related scheduling problems*, FOCS Conference proceedings, 348-356,
- [7] <http://epaedia.eea.europa.eu>, July 8, 2007,
- [8] Erlebach, T., Spieksma, F.C.R., 2003. *Interval selection: algorithms, applications, and lower bounds*, Journal of algorithms, 46 (1), 27-53,
- [9] Feuerstein, E., Stougie, L., 2001. *On-line single-server dial-a-ride problems*, Theoretical Computer Science 268 (1), 91-105.
- [10] Gupta, S.K. 1985. *Linear Programming and Network Models*, Affiliated East-West Press,
- [11] Koutsoupias, Elias, Taylor, David Scot, 2004. *The CNN problem and other k-server variants*, Theoretical Computer Science, 324, (2-3), 347-359.
- [12] Marchetti-Spaccamela, A., Vercellis, C., 1995. *Stochastic on-line knapsack problems*, Mathematical Programming, Vol 68, pp 73-104,
- [13] Otten, Bernard, 2006. *Het optimaal beladen van een vrachtwagen over een vastgestelde route*, (in Dutch), Master's Thesis, Universiteit Maastricht,
- [14] Samuelsson, Anders, Tilanus, Bernhard, 1999. *A framework efficiency model for goods transportation with an application ot regional less-than-truckload distribution*, Transport Logistics, 1 (2), 139-151,
- [15] Talluri, Kalyan T., Van Ryzin, Garrett J., 2005. *The Theory and Practice of Revenue Management*, Springer Science,
- [16] Woeginger, Gerhard, 1994. *On-line scheduling of jobs with fixed start and end times*, Theoretical Computer Science 130 (1), 5-16.